Machine Learning: Multicollinearity

Multiple Regression: Multicollinearity

- More may not be better
  - May create problems
    - Independent variable correlates with one or more independent variables
    - Independent is no longer independent!

Machine Learning: Multicollinearity

Multiple Regression: Multicollinearity

```
# VIF for Multicollinearity Testing
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

Machine Learning: Multicollinearity

Multiple Regression: Multicollinearity

- More may not be better
  - May create problems
    - Independent variable correlates with one or more independent variables
    - Independent is no longer independent!

Machine Learning: Multiple Regression Dummy Variables

Machine Learning: Multiple Regression Model

- Let's review

  $y = a + bx$

  $E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_j$

  y = dependent variable (outcome)

  x = independent variable (predictor)

Machine Learning: Multiple Regression Model

- Linear Regression
  y = continuous variable
    1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ….., ∞

- Logistic Regression
  y = categorical variable with 2 categories
    male, female

- Multinominal Regression
  y = categorical variable with more than 2 categories
    black, green, brown

In all cases, predictors (x) can be continuous or categorical

Machine Learning: Multiple Regression Model

- What happens when the predictor is a categorical variable?

$$E(y)=\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_i x_j$$

How can we use categorical data in regression analysis?

Machine Learning: Multiple Regression Model

- $E(y)=\beta_0+\beta_1 x_1$
  - For two categories, we code the data as 0 or 1

value	**X1**
Male	0
Female	1

- Two Predictive Equations
  - $\mathbf{E(y|Male)=\beta_0}$ when $x_1 = 0$
  - $\mathbf{E(y|Female)=\beta_0+\beta_1}$ when $x_1 = 1$

Machine Learning: Multiple Regression Model

- $E(y)=\beta_0+\beta_1 x_1$
  - For more than two categories, we will need to create dummy variables (transform original X_1 into dummy variables)
  - Number of dummy variables needed = # of categories - 1
- For example, we have a predictor with 4 categories
  - ART_AND_DESIGN
  - AUTO_AND_VEHICLES
  - BEAUTY
  - BOOKS_AND_REFERENCE

Machine Learning: Multiple Regression Model

- $E(y)=\beta_0+\beta_1 x_1$
  - ART_AND_DESIGN
  - AUTO_AND_VEHICLES
  - BEAUTY
  - BOOKS_AND_REFERENCE

value	**X1**	**X2**	**X3**
ART_AND_DESIGN	1	0	0
AUTO_AND_VEHICLES	0	1	0
BEAUTY	0	0	1
BOOKS_AND_REFERENCE	0	0	0

How many predictive equations?

Machine Learning: Multiple Regression Model

value	**X1**	**X2**	**X3**
ART_AND_DESIGN	1	0	0
AUTO_AND_VEHICLES	0	1	0
BEAUTY	0	0	1
BOOKS_AND_REFERENCE	0	0	0

- **E(y|Books_AND_Reference)=β_0** when x_1, x_2, $x_3 = 0$
- **E(y|Art_and Design)=$\beta_0+\beta_1$** when x_2, $x_3 = 0$
- **E(y|Auto_and_Vehicles)=$\beta_0+\beta_2$** when x_1, $x_3 = 0$
- **E(y|Beauty)=$\beta_0+\beta_3$** when x_1, $x_2 = 0$

Machine Learning: Python Create Dummy Variables

	Date	Day	Temperature	SalesClerk	Tweets	Price	Sales
0	1/1/2019	Tuesday	72	John	2	0.5	177
1	1/3/2019	Thursday	69	John	5	0.5	172
2	1/4/2019	Friday	100	John	7	0.5	150
3	1/6/2019	Sunday	91	Ada	8	0.5	120
4	1/7/2019	Monday	81	Ada	3	0.3	96

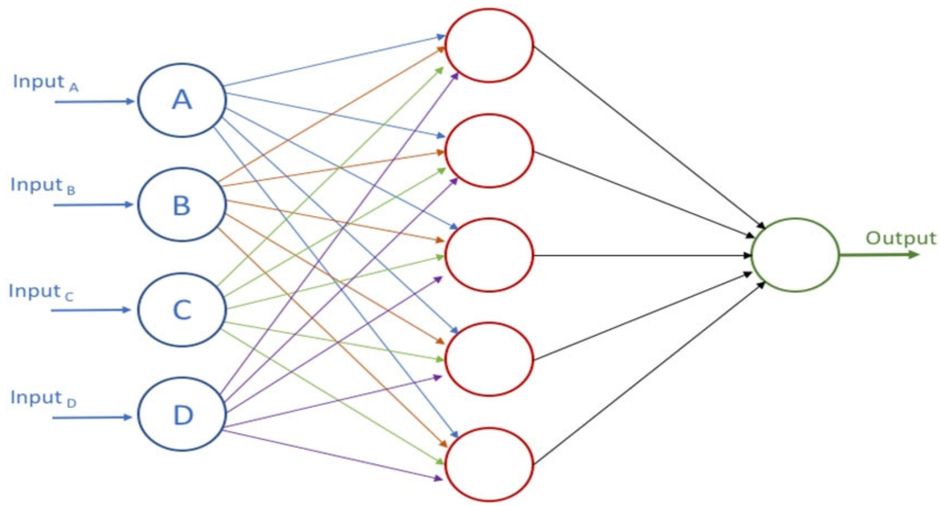
```
Dummies = pd.get_dummies(df.SalesClerk, prefix='SalesPerson',drop_first=True)

NewDF = pd.concat([df, Dummies], axis="columns")
```

Machine Learning: Python Create Dummy Variables

	Date	Day	Temperature	SalesClerk	Tweets	Price	Sales
0	1/1/2019	Tuesday	72	John	2	0.5	177
1	1/3/2019	Thursday	69	John	5	0.5	172
2	1/4/2019	Friday	100	John	7	0.5	150
3	1/6/2019	Sunday	91	Ada	8	0.5	120
4	1/7/2019	Monday	81	Ada	3	0.3	96

```
df['SalesClerk'] = df['SalesClerk'].map({'Ada':0, 'John':1})
```

Machine Learning: Classification Problems



Neural Networks

Classification

Binary (binominal)

Multi-Class (multinominal)

Machine Learning Algorithms

- Regression (Linear)
  - Predicts continuous quantity outcome
  - based on the least square estimation
  - Dependent variable: numeric
  - Independent variable: continuous numeric or categorical

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_2 X_3 + \ldots + \beta_n X_n$$

- Classification (Logistic Regression)
  - Predicts discrete categorical label
  - based on maximum likelihood estimation
  - Dependent variable: categorical
  - Independent variable: continuous numeric or categorical

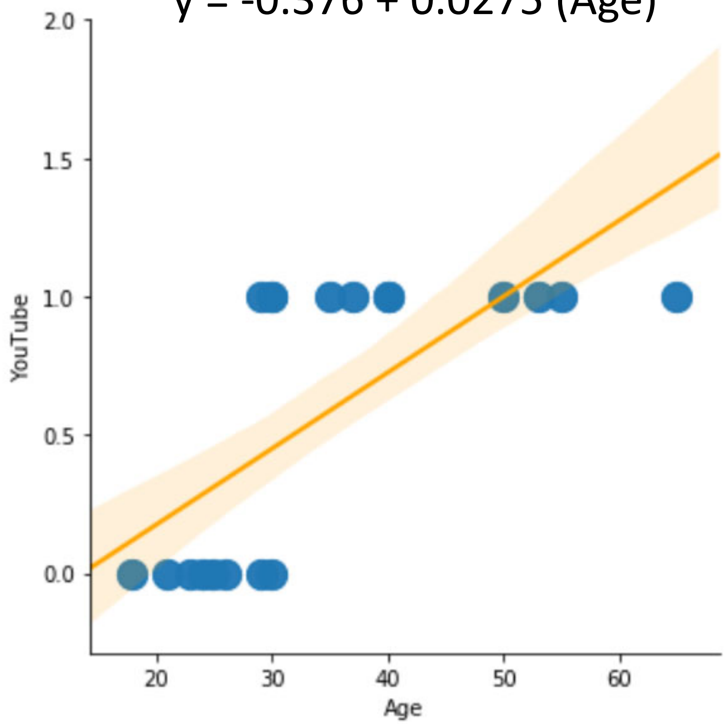
$$P = \frac{e^{(\beta_0 + \beta_1 X)}}{1 + e^{(\beta_0 + \beta_1 X)}}$$

What happened if we use linear regression on a (binary) classification problem?

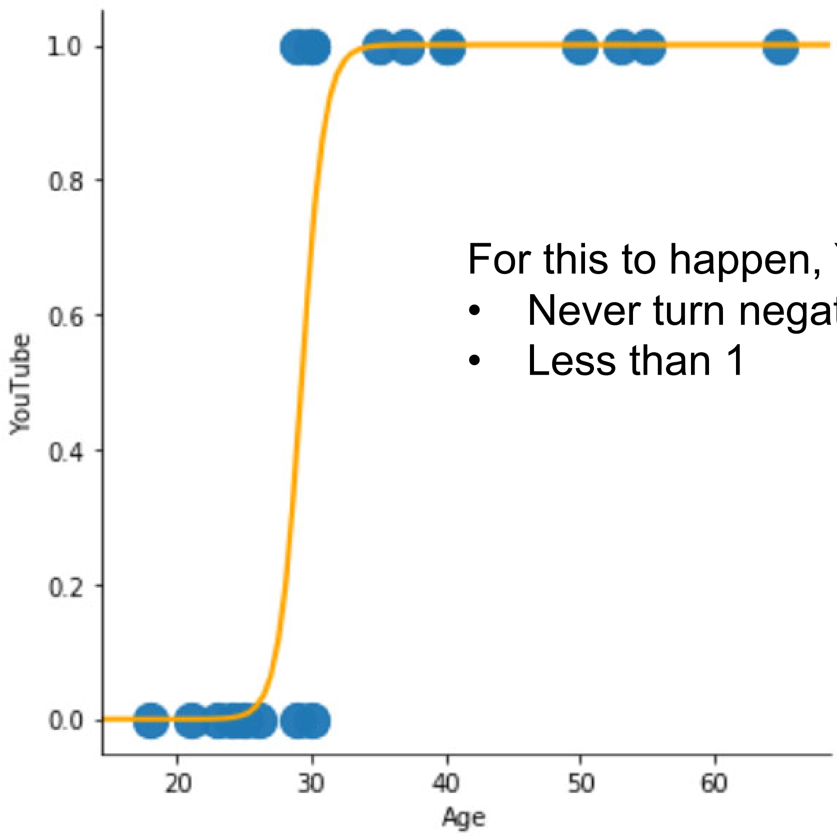
YouTube	Age
1	30
1	35
1	40
1	50
1	65
1	55
1	53
1	40
0	30
0	23
0	21
0	18
0	25
1	30
1	29
1	37
0	24
0	26
0	29
1	30
1	35
1	40
1	50
1	65
1	55
1	53
1	40
0	30
0	23

$$Y = \beta_0 + \beta_1 X$$

$$y = -0.376 + 0.0275\,(\text{Age})$$



YouTube	Age
1	30
1	35
1	40
1	50
1	65
1	55
1	53
1	40
0	30
0	23
0	21
0	18
0	25
1	30
1	29
1	37
0	24
0	26
0	29
1	30
1	35
1	40
1	50
1	65
1	55
1	53
1	40
0	30
0	23

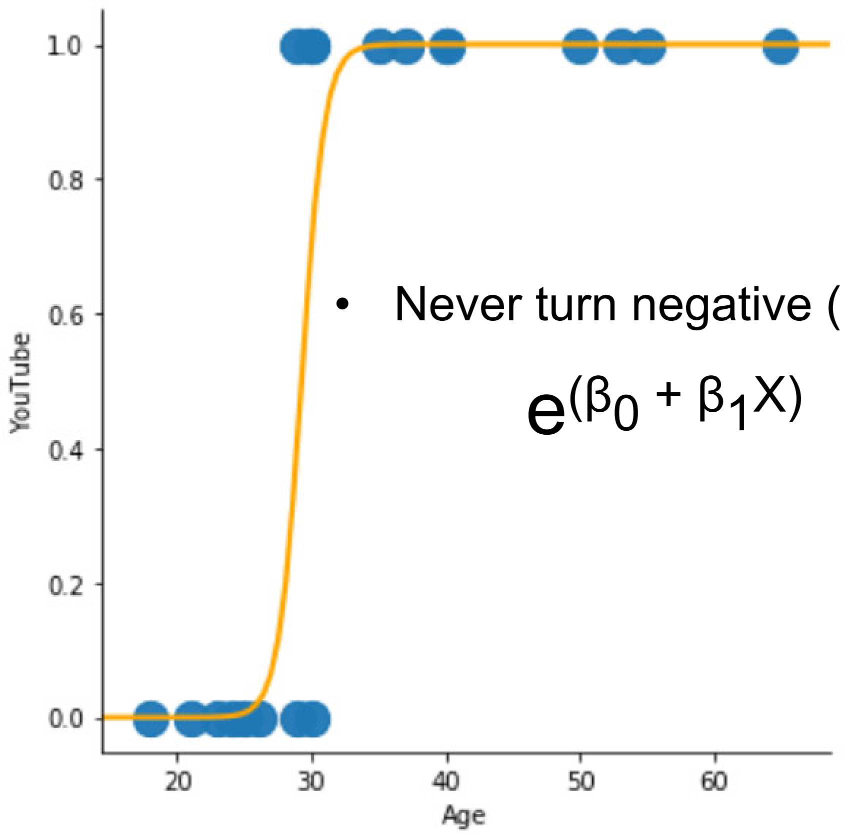


For this to happen, $Y = \beta_0 + \beta_1 X$ must:

- Never turn negative (>= 0)
- Less than 1

For this to happen, $Y = \beta_0 + \beta_1 X$ must:

YouTube	Age
1	30
1	35
1	40
1	50
1	65
1	55
1	53
1	40
0	30
0	23
0	21
0	18
0	25
1	30
1	29
1	37
0	24
0	26
0	29
1	30
1	35
1	40
1	50
1	65
1	55
1	53
1	40
0	30
0	23



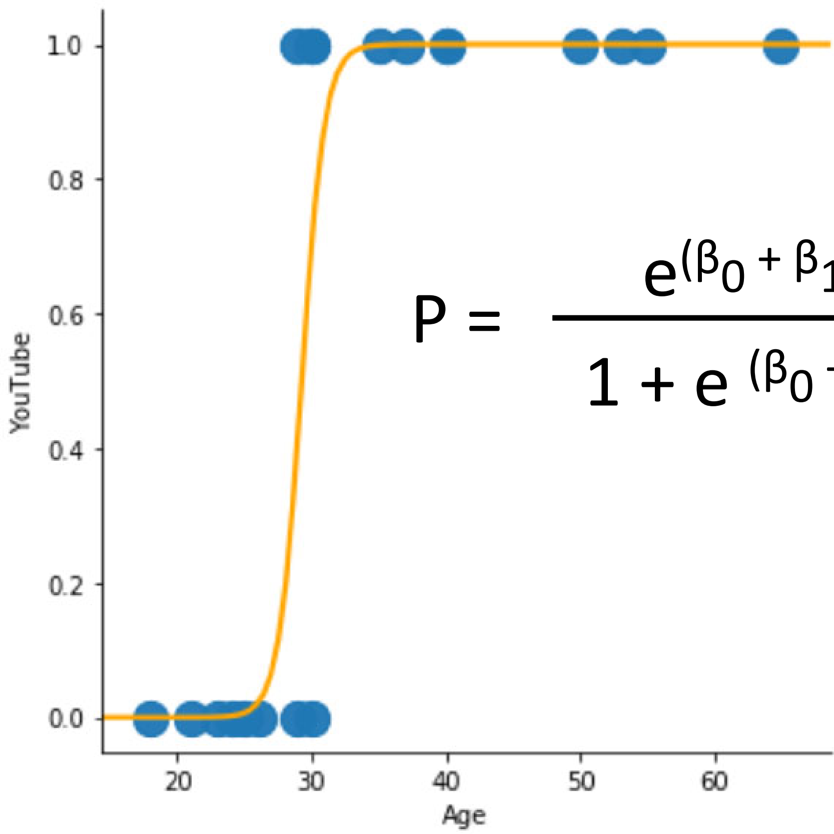
- Never turn negative (>= 0)

$$e^{(\beta_0 + \beta_1 X)}$$

- Less than 1

$$\frac{e^{(\beta_0 + \beta_1 X)}}{1 + e^{(\beta_0 + \beta_1 X)}}$$

YouTube	Age
1	30
1	35
1	40
1	50
1	65
1	55
1	53
1	40
0	30
0	23
0	21
0	18
0	25
1	30
1	29
1	37
0	24
0	26
0	29
1	30
1	35
1	40
1	50
1	65
1	55
1	53
1	40
0	30
0	23



$$P = \frac{e^{(\beta_0 + \beta_1 X)}}{1 + e^{(\beta_0 + \beta_1 X)}}$$

Machine Learning Algorithms

- Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_2 X_3 + \ldots + \beta_n X_n$$

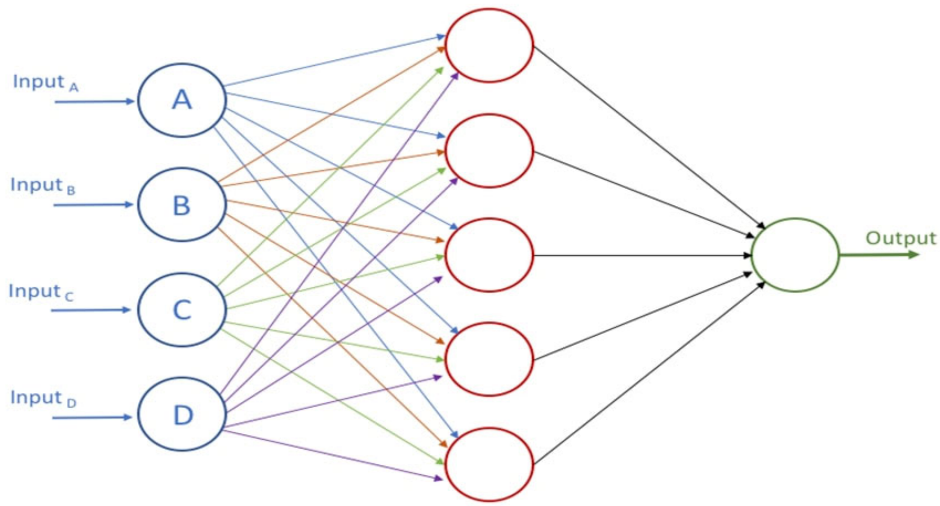
$$E(Y) = F(X)$$

- Logistic Regression

$$P = \frac{e^{(\beta_0 + \beta_1 X)}}{1 + e^{(\beta_0 + \beta_1 X)}}$$

$$E(Y) = P(Y=1)$$

Machine Learning: Classification Problems



Neural Networks

Precision – Accuracy of positive predictions

Recall: Fraction of positives that were correctly identified

A system with high recall but low precision returns many results, but most of its predicted labels are incorrect when compared to the training labels.

A system with high precision but low recall is just the opposite, returning very few results, but most of its predicted labels are correct when compared to the training labels.

An ideal system with high precision and high recall will return many results, with all results labeled correctly.

F1 score – What percent of positive predictions were correct?

Support is the number of actual occurrences of the class in the specified dataset